Gap-based mechanisms in differential privacy

Adam Smith

Penn State

Bar-Ilan Winter School
February 14, 2017

PennState
College of Engineering

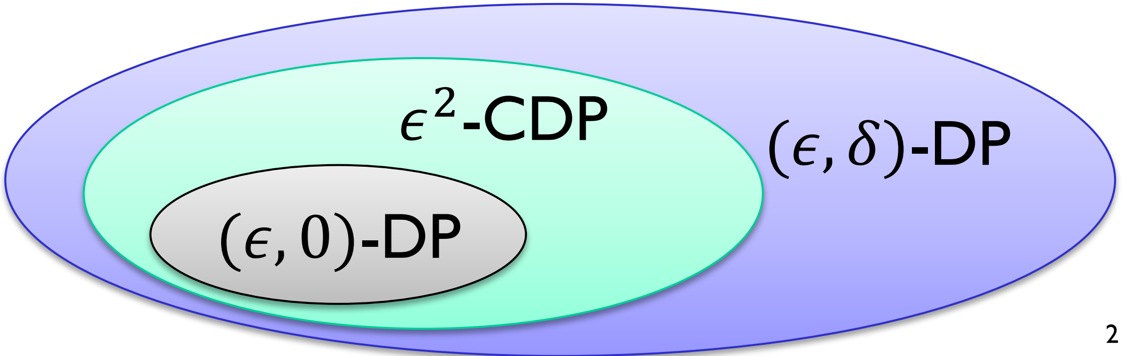
So far: global sensitivity

- Looked at releasing, or optimizing over, vector of queries $\vec{q} = (q_1, \ldots, q_k)$ with low sensitivity
  - In ℓ_1, ℓ_2 norms (noise addition)
  - In ℓ_∞ norms (algorithms for releasing counting queries)

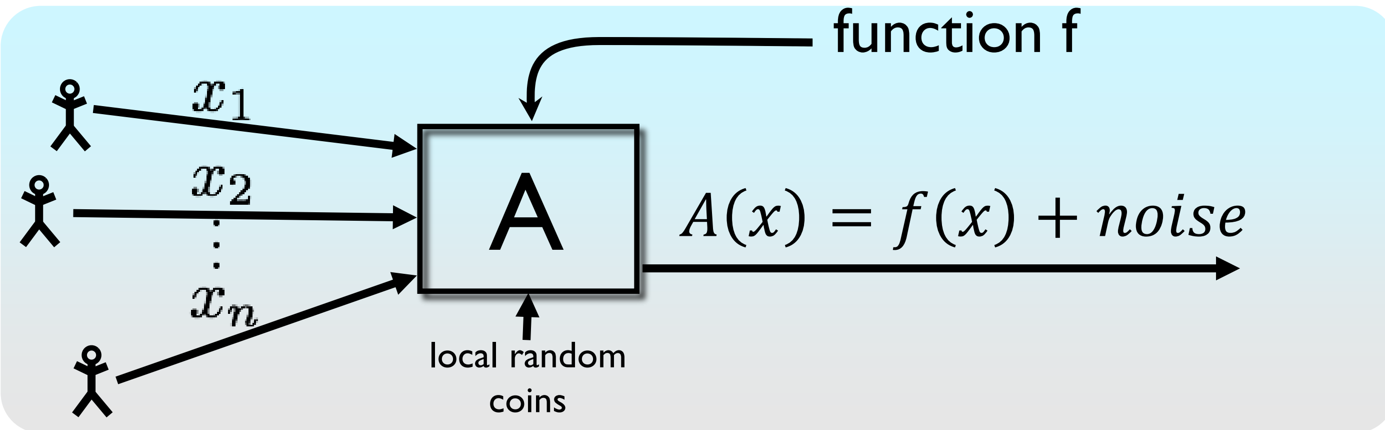
- What do we do when sensitivity is not the same everywhere?

- How can we get higher accuracy on instances where sensitivity is lower?

- Technical point

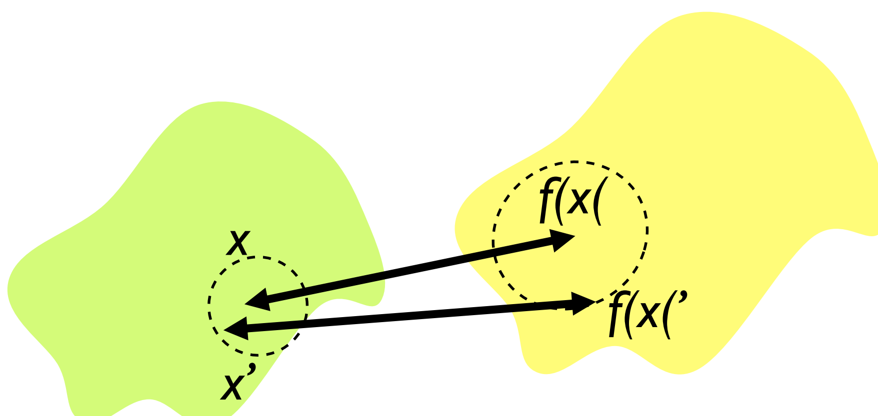


Laplace Mechanism

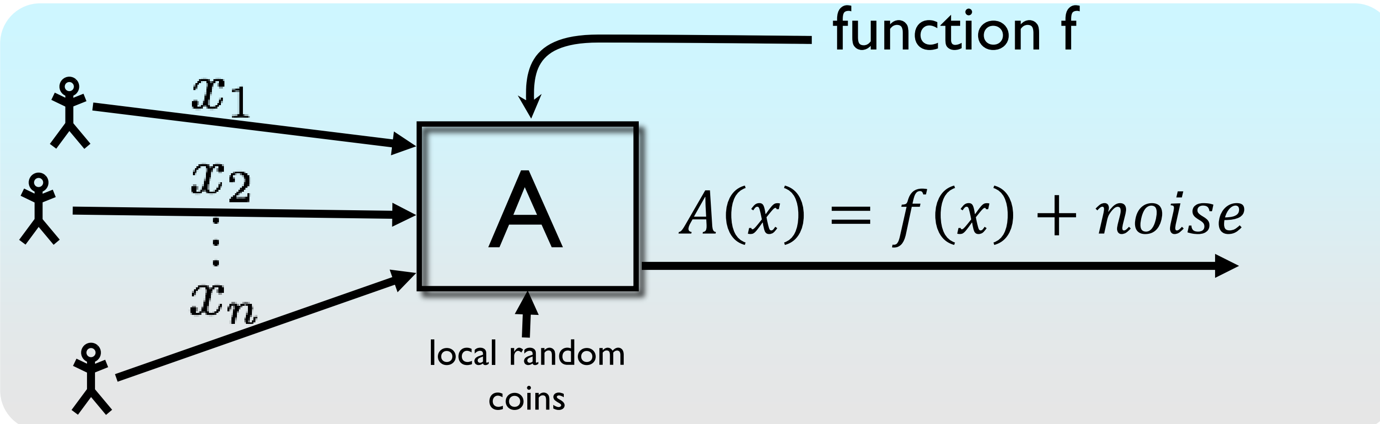


- Global Sensitivity : $\text{GS}_f = \max_{\text{neighbors } x, x'} \|f(x) - f(x')\|_1$

➢ Example $\text{GS}_{\text{proportion}} = \frac{1}{n}$



Laplace Mechanism



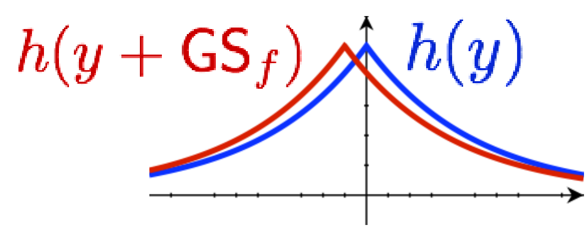
- Global Sensitivity: $\mathrm{GS}_f = \max_{\text{neighbors } x, x'} \|f(x) - f(x')\|_1$
  - Example: $\mathrm{GS}_{\text{proportion}} = \frac{1}{n}$

Theorem: If $A(x) = f(x) + \mathrm{Lap}\left(\frac{\mathrm{GS}_f}{\epsilon}\right)$, then A is ϵ-differentially private.

- Laplace distribution $\mathrm{Lap}(\lambda)$ has density

$$h(y) \propto e^{-|y|/\lambda}$$

- Changing one point translates curve



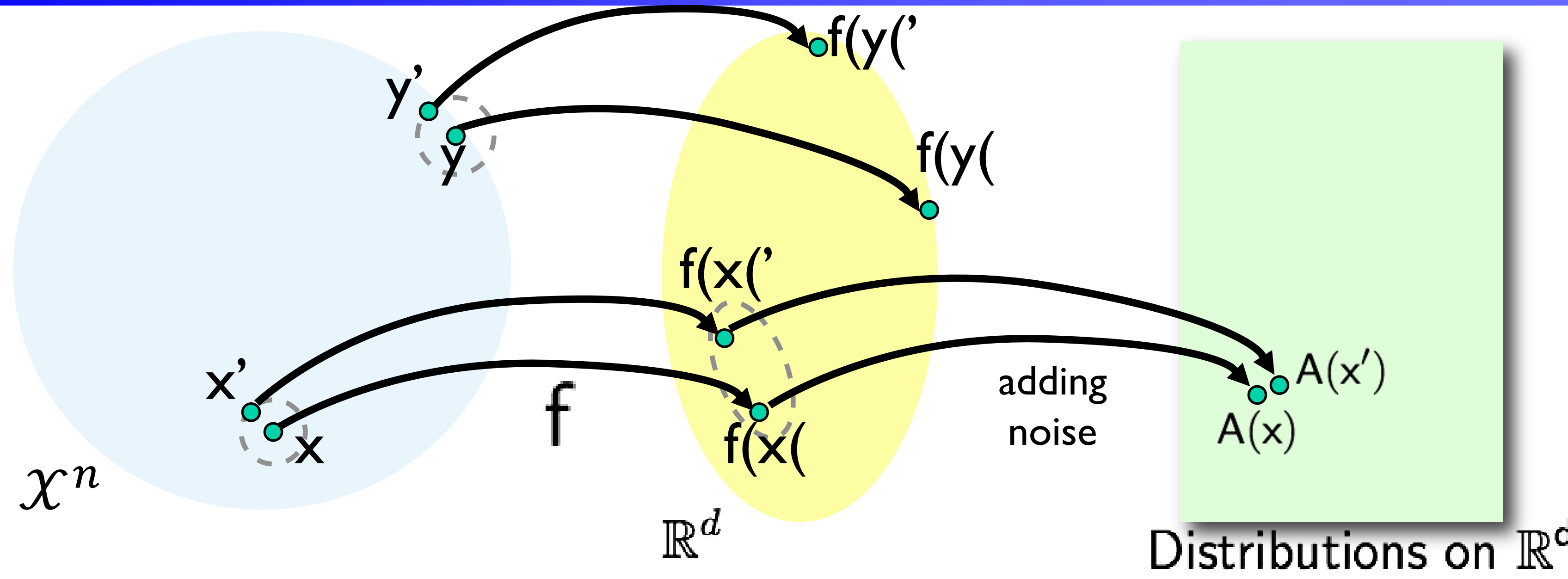
Variants in other metrics

- Consider f : $\mathcal{D}^n \rightarrow \mathbb{R}^d$
- Global Sensitivity: $GS_f = \max_{\text{neighbors } x,x'} \| f(x) - f(x') \|_2$

Theorem: If A(x) = f(x) + $N\left(0, \left(\frac{2GS_f\sqrt{\ln(1/\delta)}}{\epsilon}\right)^2\right)$, then A is (ϵ, δ)-differentially private.

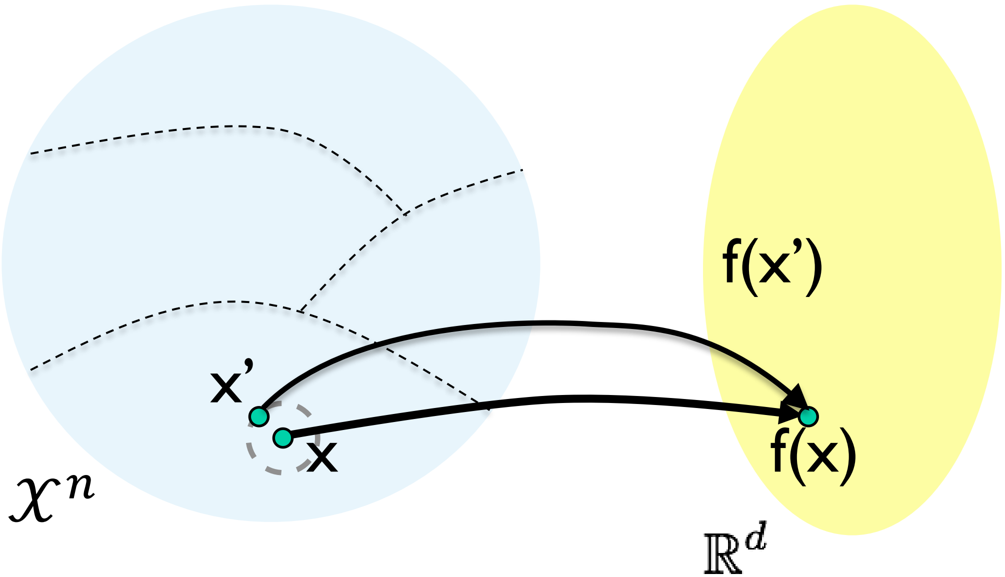
- Example ... cates
  - f(x) = vector of counts.
  - $GS_f = \sqrt{d}$
  - Add noise $\frac{\sqrt{d \ln(1/\delta)}}{\epsilon}$ per entry instead of $\frac{d}{\epsilon}$.
  - Also possible with Laplace noise and strong composition
  - Tight by "membership testing" attacks [BUV]

Global versus local [NRS07]

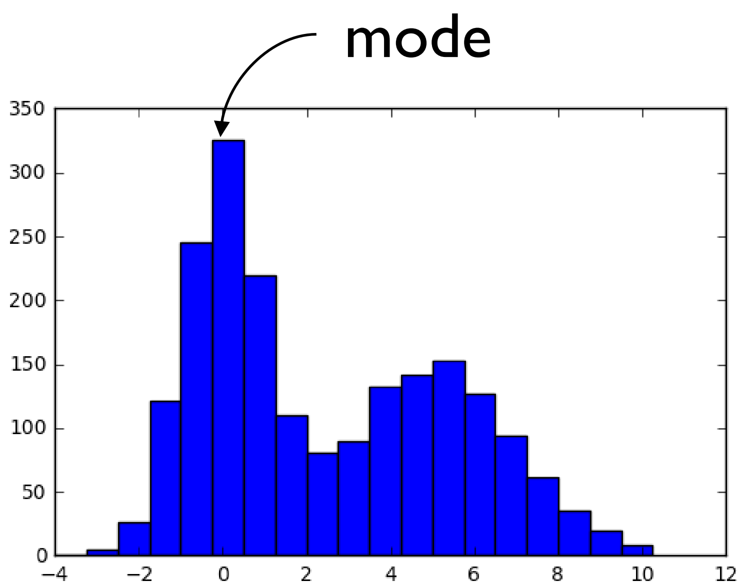


- Global sensitivity is worst case over inputs
- Local sensitivity: $$\mathsf{LS}_f(x) = \max_{x' \text{ neighbor of } x} \| f(x) - f(x') \|_1$$
- Reminder: $\mathsf{GS}_f(x) = \max_x \mathsf{LS}_f(x)$
- [NRS’07,DL’09, …] Techniques with error ≈ local sensitivity
  - ➢ Basis of best algorithms for graph data

Extreme case: piece-wise constant functions

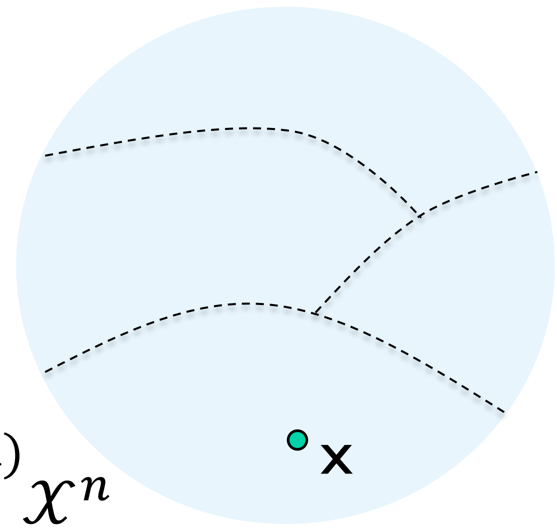


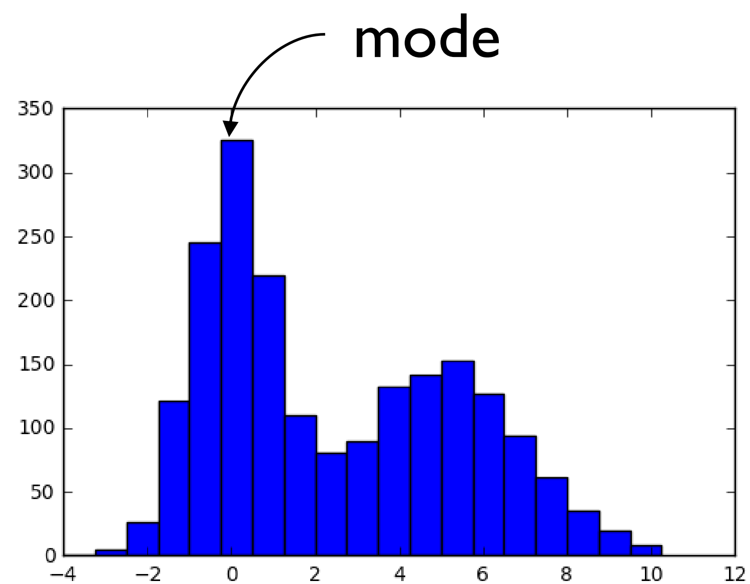
- Consider $mode$ function:
  - given $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$, return the most frequent value (breaking ties lexicographically)
  - Pre-images are contiguous in Hamming space



Extreme case: piece-wise constant functions

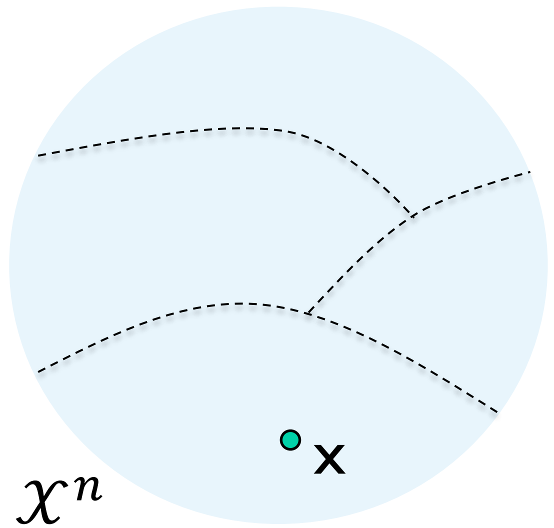
- What is the sensitivity of mode?
  - ➢ Can we "add" noise to mode?
- DP Mechanisms?
  - ➢ Release the entire histogram
    - Noise $1/\epsilon$ per entry
    - Report $\widehat{mode} = argmax(noisy\ histogram)$
  - ➢ Use reportNoisyMax (exponential mechanism)?
    - Same as previous option
- **Lemma:** $count\left(\widehat{mode}\right) \geq count(mode) - O\left(\frac{\log(d)}{\epsilon}\right)$ w.p. $1 - o(1)$.
- Can we avoid dependency on $|\mathcal{X}|$?
  - ➢ (Not in worst case)



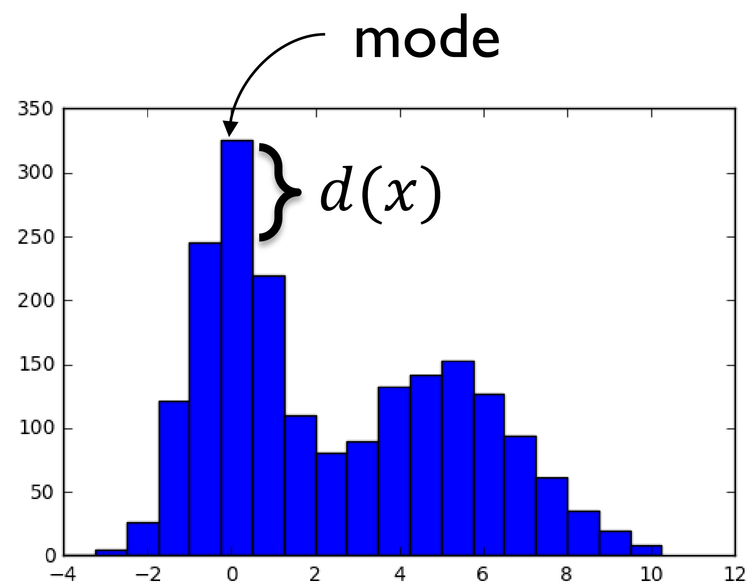


Extreme case: piece-wise constant functions

- What is the sensitivity of mode?
  - Can we "add" noise to mode?

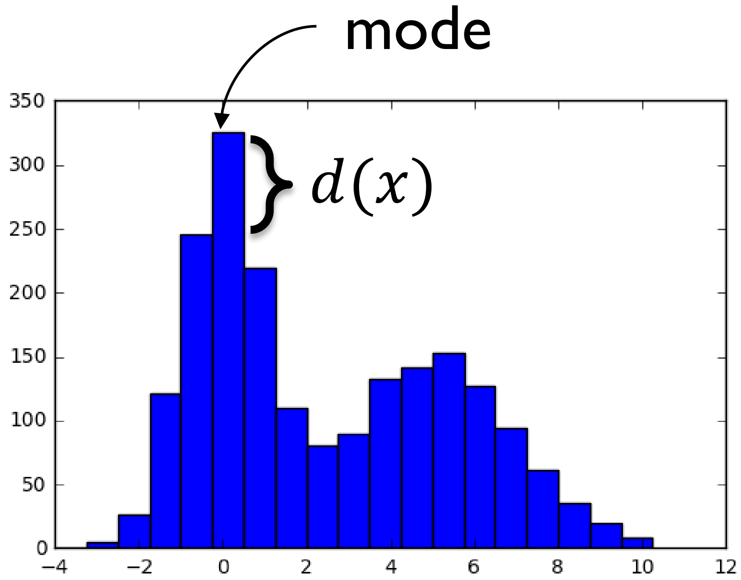


- How far am I from the nearest data set with a different mode?
  - $dist(x) = \max(x) - secondmax(x)$
- How sensitive is GS_f?
  - $GS_{dist} = 1$
  - We can release $dist(x) + Lap\left(\frac{1}{\epsilon}\right)$



Stability-based mode

- $dist(x)$
  = Hamming distance to nearest database with a different mode
  $= \max(x) - secondmax(x)$



- $A_{dist}(x)$:
  - $\tilde{D} = dist(x) + Lap\left(\frac{1}{\epsilon}\right)$
  - If $\tilde{D} > \ln\left(\frac{1}{\delta}\right)/\epsilon$:
    - Return $\tilde{D}$ and exact $mode(x)$
  - Else:
    - Return $\tilde{D}$ and $\perp$
- **Proposition**: $A_{dist}(x)$ is (ϵ, δ)-DP
- **Proposition**: If $dist(x) > t/\epsilon$, then

$$\Pr(A_{dist}(x)\ releases\ mode) \geq 1 - e^{-t}$$

Proposition: $A_{dist}(x)$ ***is*** (ϵ, δ)***-DP***

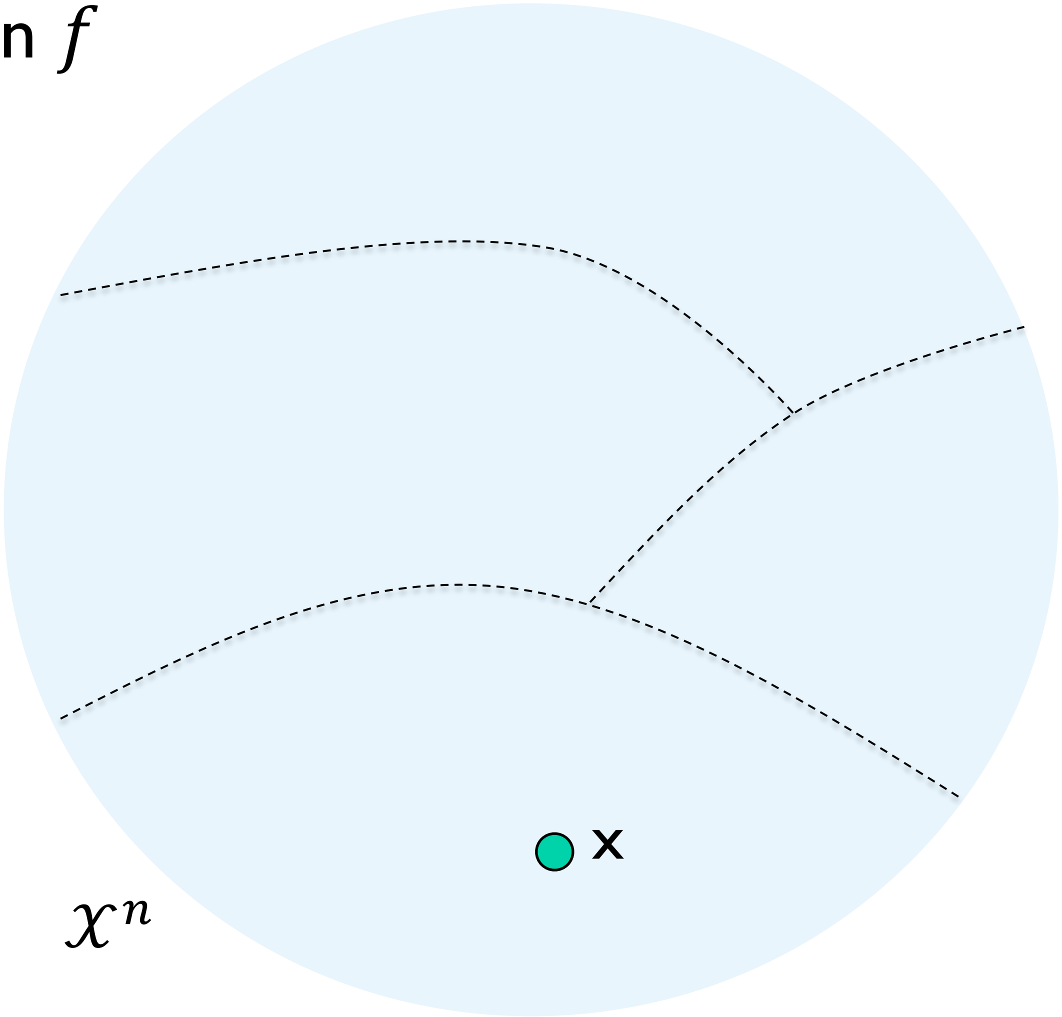
Proposition: If $dist(x) > t/\epsilon$***, then***

$$\Pr(A_{dist}(x)\ releases\ mode) \geq\ 1 - e^{-t}$$

Stability mechanism

- Works for any function f

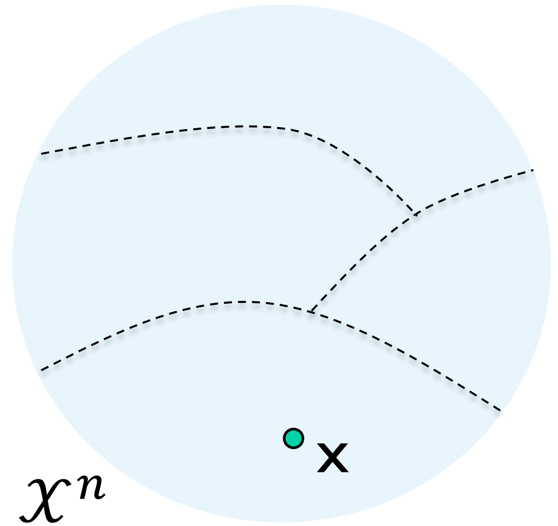
- We can release $f(x)$ when x is far from input with different answer
  - Regardless of domain size



Propose-Test-Release [Dwork, Lei 2009]

General principle: Let

- B be an algorithm that satisfies (ϵ, δ)-DP on a subset $Y \subseteq \mathcal{X}^n$ of data sets
  - Specifically, for all neighboring data sets $x, x' \in Y$, $\Pr(B(x) \in T) \leq e^{\epsilon}\Pr(B(x') \in T) + \delta \quad (\forall T \subseteq R)$
- $dist_Y(x)$ = Hamming distance to complement of Y
- $A_{Y,B}(x)$:
  - $\tilde{D} = dist_Y(x) + Lap\left(\frac{1}{\epsilon}\right)$
  - If $\tilde{D} > \ln\left(\frac{1}{\delta}\right)/\epsilon$:
    - Return $\tilde{D}$ and $B(x)$
  - Else:
    - Return $\tilde{D}$ and $\perp$



Application: Sampling stability [S, Thakurta13]

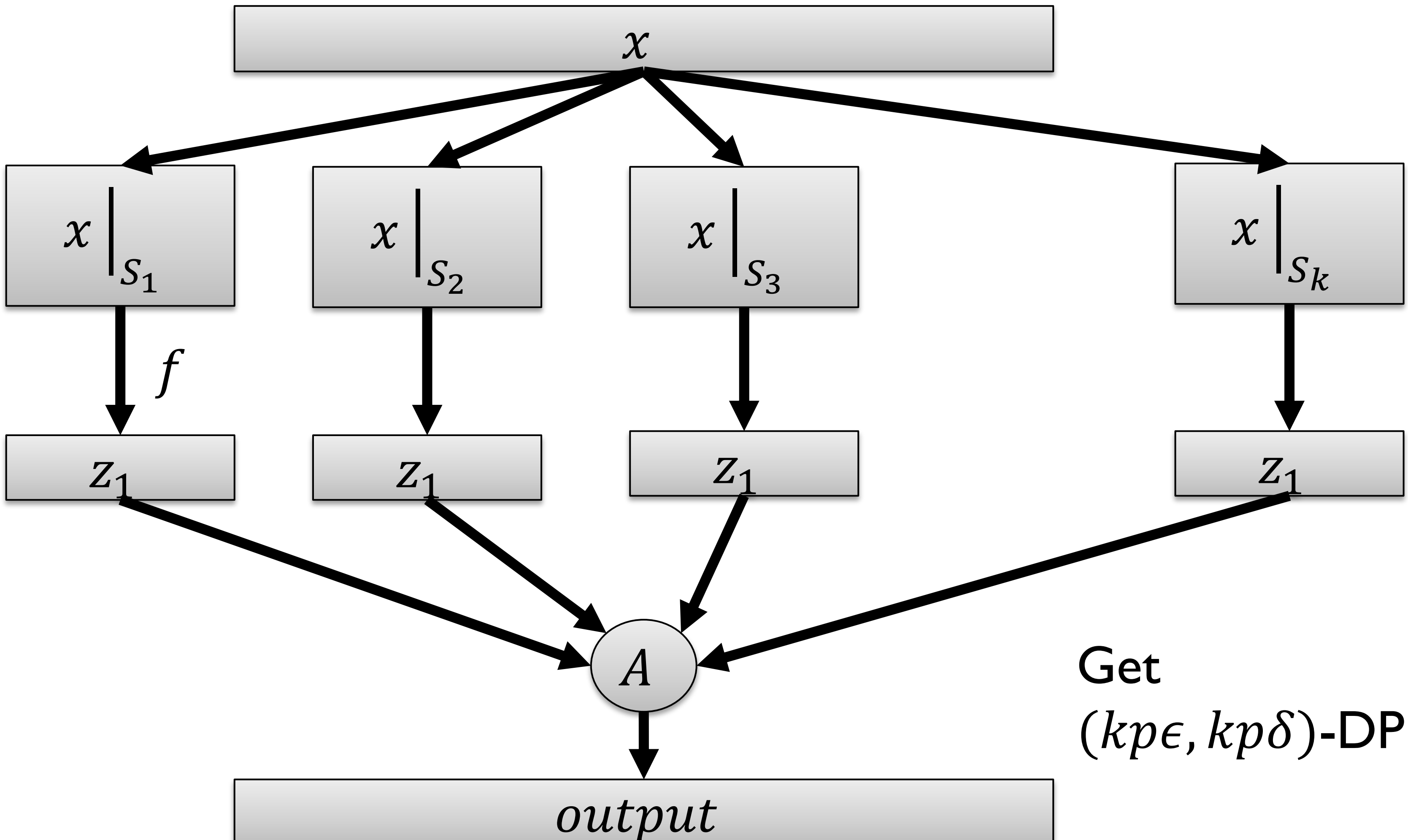
- Common computational technique to achieve robustness/stability:
  - compute a function on random subsamples from input
- Given $p \in [0,1]$, a p-subsample from x is a uniformly random subset of x of size pn
- **Definition:** an function f is p-subsampling-stable on x if there exists a value z^* such that

$$\Pr_{S:|S|=pn}\left(f\left(x\Big|_S\right) = z^*\right) \geq \frac{2}{3}$$

- How can we exploit this type of stability?

Subsample and aggregate [Nissim Raskhodnikova S 2007]

- For any $(\epsilon, \delta) - DP$ algorithm A:



Get $(kp\epsilon, kp\delta)$-DP

Sample and aggregate

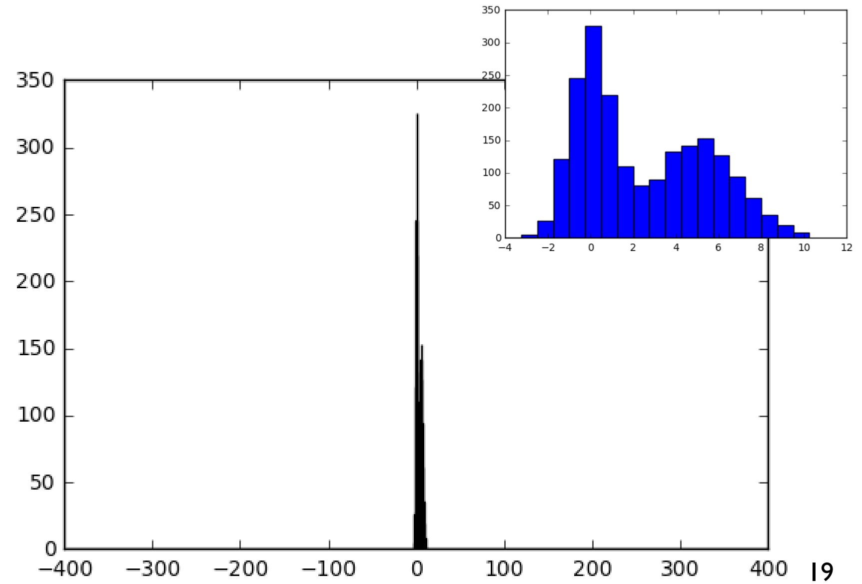
- **Definition:** an function f is p-subsampling-stable on x if there exists a value z^* such that

$$\Pr_{S:|S|=pn}\left(f\left(x\Big|_S\right) = z^*\right) \geq \frac{2}{3}$$

- Sample and aggregate: if $p < \epsilon t$, then sample and aggregate with stable mode returns z^* with probability $\geq 1 - e^{-t}$

Getting the whole histogram

- Say we want to release a histogram of data from a huge domain.
  - Not just the mode!
  - Want counts of all bins n_j for $j = 1, \ldots, d$
- First attempt: just add noise
  - $A(x)$: For all bins: $\tilde{n}_j = n_j + Lap\left(\frac{1}{\epsilon}\right)$
  - Problem: huge output!
- Return only $\{(j, \tilde{n}_j): \quad \tilde{n}_j > \tau\}$
  - Problem: if domain is large, many spurious bins!
  - If $\log(d) \gg n$, then get more noise than signal

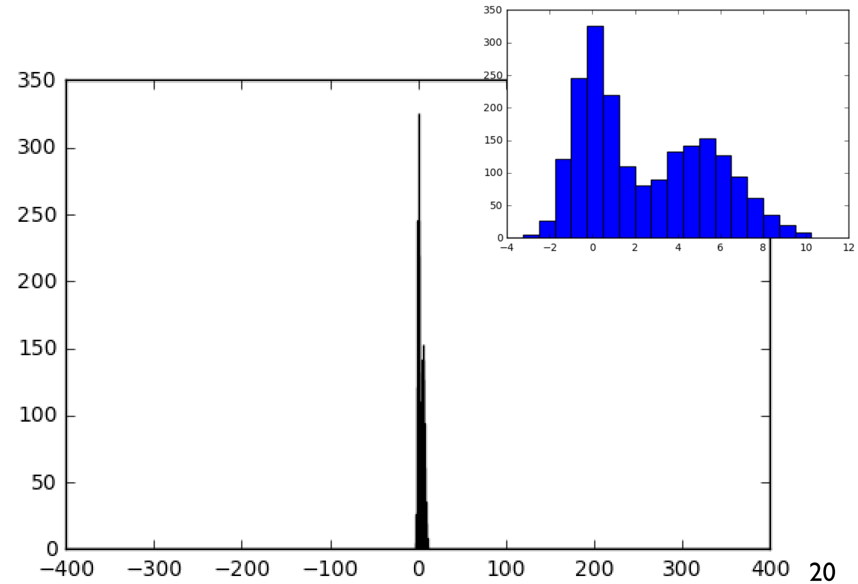


Truncated Histrogram

- $A(x)$:
  - For all bins with nonzero counts: $\tilde{n}_j = n_j + Lap\left(\frac{1}{\epsilon}\right)$
  - Return only $\{(j, \tilde{n}_j): \ \tilde{n}_j > \tau\}$ with $\tau = \ln\left(\frac{1}{\delta}\right)/\epsilon$

- **Prop:** A is (ϵ, δ)-DP

- **Prop:** With prob $1 - e^{-t}$, A returns all bins with $n_j \geq \frac{\ln(n)}{\epsilon} t$



Separating definitions

- Gap-based histogram shows that (ϵ, δ)-DP algorithms can have

$$I\big(X; A(X)\big) \approx \epsilon n log(d)$$

  - Unbounded!
  - Requires very different proof mechanisms

Gap-based mechanisms

- These ideas applied to a variety of problems
  - “Exponential mechanism with gaps”
  - Learning point functions
  - Releasing ”robust” statistics

- Basic idea: look for conditions under which the output is stable, test for those conditions